

Amendments to the Specification:

Please replace the paragraph on page 7, beginning on line 16 with the following amended paragraph:

Figure 2 is a block diagram of a system 200 in testing unit 105 according to a specific embodiment of the invention. A driver 205 can generate packets based on patterns 210, where a single pattern can be responsible for generating a plurality of packets. A network interface 215 permits packets to be transmitted to and received from the network 110. Thus, network traffic 220 can be transmitted by and received by the network interface 215.

Please replace the paragraph on page 21, beginning on line 13 with the following amended paragraph:

(II) Value Injection - Value injection does not alter the grammar of a transaction. Instead, as shown in Figure 5, value injection injects or alters the input fields 505 in a transaction 500. Value injection can be useful, for example, to exercise inputs to Application Programming Interface (API) calls and to change the behavior of database transactions. The value injection techniques may be combined with grammar injection to obtain maximum effect.

Please replace the paragraph on page 36, beginning on line 1 with the following amended paragraph:

As shown in Figure 9, another method is by having a translator 900 convert a delimiter into a character that should not be passed to the shell. For example, as shown in Table 19, assume that the original transaction content 902a includes primary command 905 ("username"), delimiter 911 ("%3F"), and additional command 915 ("rm -rf/"). The parser will pass the original content "username%3Frm -rf/". However, the translator 900 will convert the original delimiter "%3F" into ";". As a result, the parser 830 will parse the translated transaction 902b with content "username;rm -rf/", leading to a fault event 910 of hard drive content deletion.

Please replace the paragraph on page 38, beginning on line 15 with the following amended paragraph:

Figure 10 illustrates a block diagram of a method of using extraneous meta-characters for causing fault injection in an SUT (or target) 1000. A parsing event (performed by parser 1005) acts upon an input transaction 1010. Based on the results of the parsing event, a result (A) 1025 or result (B) 1020 is generated. The input transaction 1010 can be changed such that, after the parsing event, a result (A) 1025 is changed into a result (B) 1020 (or a result (B) 1020 is changed into a result (A) 1025). If, for example, the input transaction 1010 is changed so that the parsing event generates the result (B) 1020 (instead of a result (A) 1025), then particular problems (such as a security violation) will occur in the SUT 1000 since the SUT will now be operating on data that it is not supposed to see.

Please replace the paragraph on page 65, beginning on line 1 with the following amended paragraph:

Figure 12 is a state diagram 1200 illustrating a fault detection method according to an embodiment of the invention. In state 1205, the target system (i.e., system-under-test) waits for a transaction from the network. The transaction may be sent across the network by, for example, the testing unit 105 (Figure 1). In state 1210, parsing is performed on the received transaction. Based on the results of the parsing in state 1210, a fault condition (e.g., a program crash) may occur as shown in state 1215, or a non-fault condition may occur as shown in state 1220. If the fault condition of state 1215 occurs, then the target system may perform a reset as shown in state 1225. The testing unit 105 can then detect the reset occurrence in state 1225 to detect the fault occurrence.